TABLE OF CONTENTS

	Page
Question Presented	2
Interest of Amicus Curiae	3
Nature of the Software Sub-Industry	5
Summary of Argument	12
Argument	14
I. Software as a Machine Device Is Patentable Equally with Hardware	14
II. The Patent Office Search and Classification Facilities Meet the Same Standards of Adequacy for Software as Those for the Equivalent Hardware	17
III. Copyrights and Legislation for Other Forms of Legal Protection of Proprietary Rights Not Effective to Protect the Machine Devices of Software	18
1. Other Legislation	18
2. Copyrights	19
IV. The Finding of Fact That Software Is a Machine Device and Patentable Will Encourage Innovation and Competition in the Computer Industry and Will Assist in Eliminating Antitrust Violations and Their Anti-Competitive Effects	21
Conclusion	25
Appendices:	
1. IBM U.S. Patent No. 3,624,637, issued November 30, 1971	A-1
2. Honeywell U.S. Patent No. 3,505,675, issued April 7, 1970	A-6

TABLE OF CONTENTS—*(Continued)*

Page

3. Editorial in Computer Decisions, March 1972, p. 48 ..... A-13

4. Portions of Consent Decree in U.S. v. IBM, 1956 Trade Cas., para. 68245, Secs. II(a), (c), (d), (e), (f), (g), XV(b) (S.D.N.Y. 1956) A-14

TABLE OF AUTHORITIES

Cases:

Advanced Business System v. SCM, 415 F.2d 55 (C.A. 4, 1969) 22

ADR v. IBM, Civ. No. 158 (D. Minn., filed March 1969) 5, 6, 7, 8

Ex parte King and Barton (1964) 1946 USPQ 590 ... 16

Ex parte Lewin (1966) 154 USPQ 487 16

Fortner Enterprises v. U.S. Steel, 394 U.S. 495, (1969) 22

Graham v. Deere (1966) 383 U.S. 1 20

Hotchkiss (1850) 11 How. 284 20

In re Benson, 441 F.2d 682 (1971) 6, 12, 17

In re Bernhart, 417 F.2d 1395 (1969) 6, 16, 21, 24

In re Foster, 438 F.2d 1011 (1971) 6

In re Mahony, 421 F.2d 742 (1970) 6

In re Musgrave, 431 F.2d 882 (1970) 6

In re Prater, 415 F.2d 1393 (1969) 6, 16, 21, 24

In re Waldbaum, 173 USPQ 430 (1972) 6

International Salt Co. v. U.S., 332 U.S. 392 (1947) ... 24

Mazur v. Stein, 98 L. Ed. 630, 642 20

TABLE OF AUTHORITIES—*(Continued)*

Cases: Page

No. Pacif. Ry. v. U.S., 356 U.S. 1 (1958) 22

Programmatics, Inc. v. IBM, Civ. No. 159 (D. Minn. 1969) 6, 7, 21

Sears v. Stiffel (1964) 374 U.S. 826 20

Times Picayune v. U.S., 345 U.S. 594 (1953) 22

U.S. v. IBM, 298 U.S. 131 (1935) 22, 23, 24

U.S. v. IBM, 1956 Trade Cas., par. 68245 (Consent Decree) 13, 23

U.S. v. IBM, Civ. No. 69-299 (S.D.N.Y.) filed Jan. 1969 14

Washington Gas Light Co. v. Virginia El. & P. Co., 1970 Trade Cas., para. 73073 (E.D. Va. 1970) .. 22

Statutes:

15 USC 1, Sherman Act, Section 1 ..... 12, 22, 23

15 USC 14, Clayton Act, Section 3 ..... 12, 22, 23

MISCELLANEOUS

S. 1042, 90th Congress (1967) 18, 19

H. R. 5924, 90th Congress (1967) 18, 19

Automatic Data Processing Glossary, Executive Office of the President, Bureau of the Budget, Dec. 1962 2

Bowden, The Language of Computers, 58 American Scientist 43, 50, Jan.-Feb. 1970 ..... 9, 17

Brenner, Commissioner of Patents, The Future of Computer Programs in the United States Patent Office, Law of Software, 1968, pp. B-1, 12 17

MISCELLANEOUS—*(Continued)*

Page

Brooks Subcommittee of the Committee on Government Operations, H. R., 92nd Congress, May 20, 1971, Testimony of D. A. Ink, Office of Management and Budget 6

Brooks Subcommittee, Report on H. R. 4845, Aug. 17, 1965, 89th Congress, 1st Sess., House Report No. 802 7

Computer Decisions, March 1972, Editorial, p. 48 ... 8

Davis, Dr. Ruth M., Statement before the Government Activities Subcommittee, House Committee on Government Operations, May 16, 1972 at p. 12 ..... 8, 9, 10

Galbi, Proposal for New Legislation to Protect Computer Programming, 17 Bulletin Copyright Society of the U.S.A. 20

Gorn, Prof. Saul, Final Report No. AD 59 URI from Univ. of Penna. Moore School of Elec. Eng. to U.S. Army Signal Research & Development Lab. 15

Hearings on H. R. 5924 Before Subcommittee No. 3 of H. R. Committee on the Judiciary 19

Honeywell Patent U.S. 3,505,675 ..... 4, 16

IBM Patent U.S. 3,624,637 ..... 4, 16, 19

Jacobs, *Computer Technology (Hardware and Software): Some Legal* Implications for Antitrust, Copyrights and Patents, Rutgers Journal of Computers and Law, Vol. 1970, No. 8, at p. 50 . 2, 19, 20

Jacobs, M., Testimony in Hearings on S. 597 Copyright Bill, Before the Subcommittee on Patents, Trademarks and Copyrights of the Senate Committee on the Judiciary, 90th Congress, 1st Session, Pt. 3 ..... 10, 20

MISCELLANEOUS—*(Continued)*

Page

Kushner, Dr. Lawrence M., Statement Before the Government Activities Subcommittee House Committee on Government Operations, May 16, 1972 10

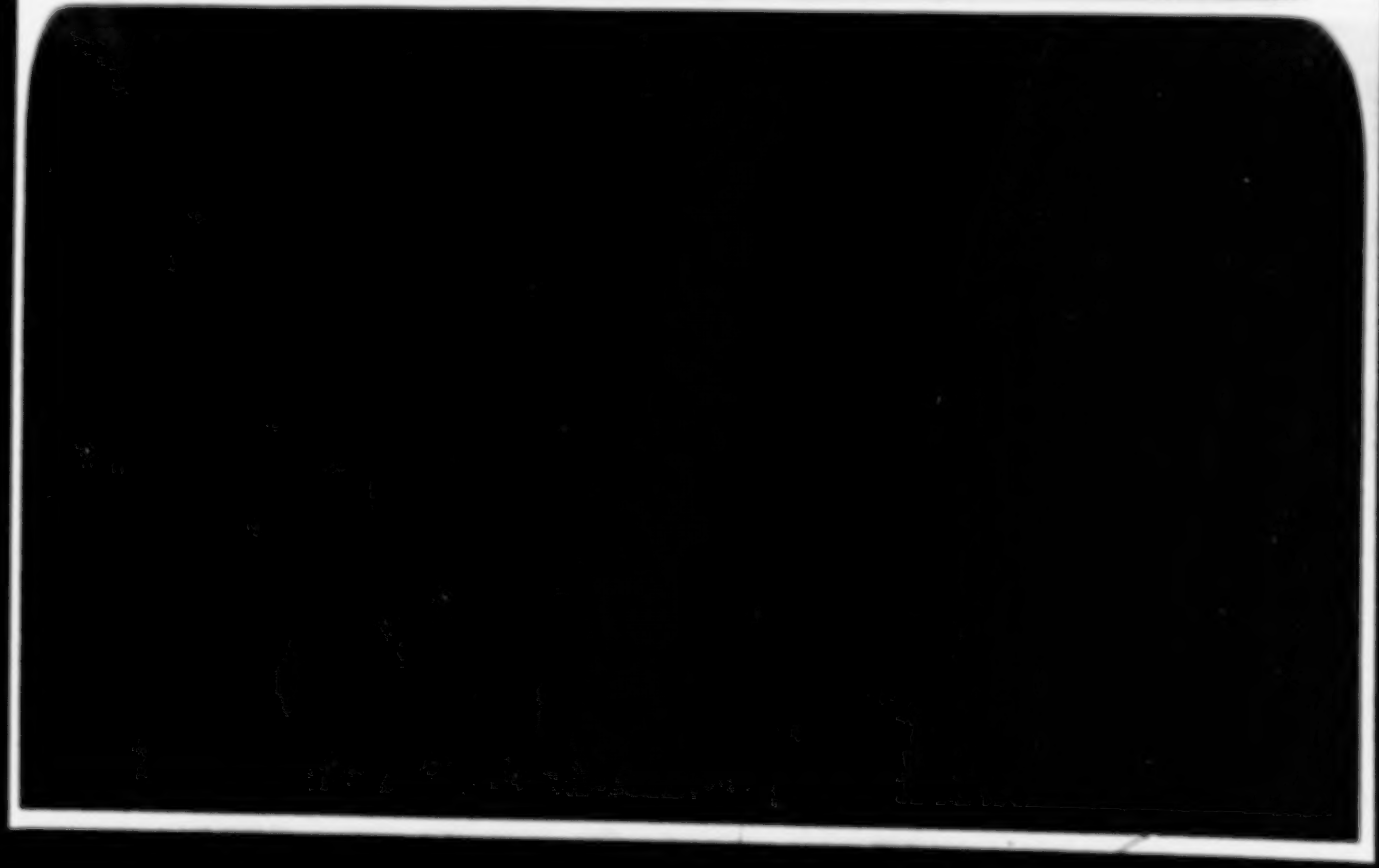
National Bureau of Standards Experimental Technology Incentives Program, Preliminary Program Description, Fiscal Year 1973, Budget Submission 9

Rosen, Prof. Saul, Hardware Design Reflecting Software Requirements, 33 Proc. of the Fall Joint Computer Conference, Dec. 1968, Pt. II, pp. 1443 ..... 2, 11, 14, 15

Software, International Data Corp., Feb. 28, 1972 ..... 7, 8, 11

Sprague, Richard, Letter to Judge Philip Neville in U.S. v. IBM and Control Data v. IBM, published in Datamation, July 1972, p. 134 7

President's Commission on Patents, Report 18



IN THE

Supreme Court of the United States

October Term 1971

No. 71-485

ROBERT GOTTSCHALK, COMMISSIONER OF PATENTS,
Petitioner

v.

GARY R. BENSON and ARTHUR C. TABBOT,
Respondents

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

BRIEF AMICUS CURIAE FOR THE ASSOCIATION OF DATA PROCESSING SERVICE ORGANIZATIONS, SOFTWARE PRODUCTS AND SERVICE SECTION (ADAPSO/AISC)

The Association of Data Processing Service Organizations, Software Products and Services Section (hereinafter ADAPSO/AISC) files this brief amicus curiae in opposition to Petitioner, and also files letters of consent of the parties in accordance with the Rules of the Court.

QUESTION PRESENTED

Whether
an automatic data processing machine or method
implemented by software[1] programming
is patentable equally with
the same machine or method
implemented by hardware[2] programming.[3]

---

1. "software, the totality of programs and routines used to extend the capabilities of computers, such as compilers, assemblers, narrators, routines and subroutines" (Automatic Data Processing Glossary, Executive Office of the President, Bureau of the Budget, . . . "for use as an authoritative reference by all officials and employees of the executive branch of the Government," Dec. 1962, p. 50).

2. "hardware, the physical equipment or devices forming a computer and peripheral equipment" (Ibid. p. 25).

For a discussion of relationships of the technologies of hardware and software as they bear on legal issues before the Court, see Jacobs, *Computer Technology (Hardware and Software): Some Legal Implications for Antitrust, Copyrights and Patents*, Rutgers Journal of Computers and Law, Vol. 1970, No. 8, at p. 50.

3. Programming built into hardware; see "Hardware Design Reflecting Software Requirements," by Prof. Saul Rosen, 33 Proc. of the Fall Joint Computer Conference, Dec. 1968, Pt. II, pp. 1443, 1444.

INTEREST OF AMICUS CURIAE

ADAPSO/AISC is a trade association[4] of companies in the software sub-industry of the computer industry. Its members are variously engaged in the marketing of software products and software services to computer users and to hardware manufacturers.

ADAPSO/AISC presents its views to the Court because a distorted description of the software technology and the software sub-industry has been set forth by Petitioner's Brief and the supporting briefs of the hardware manufacturers and their association, BEMA. Moreover, with total lack of candor, the hardware manufacturers have failed to advise the Court that they have often obtained U.S. patents for hardware-implemented

---

4. ADAPSO/AISC was formed by merger in June 1972 of the Software Products and Service Section of ADAPSO and the Association of Independent Software Companies (AISC), the latter comprising 8 companies and the former 25 companies. In a recent poll of the members, taken in connection with the filing of this brief amicus curiae (75% responding), more than 60% were classified as small business, and 80% had annual sales under $10 million, 20% have applied for a software patent, 15% have such a patent, 60% expect to apply for such a patent within the next five years, 35% would be adversely influenced by a lack of patent protection in making a decision on whether to invest in research and development in software technology. Members of ADAPSO/AISC are:

American Information Development, Applied Data Research, Atlantic Software, Inc., Bank Computer Network Corp., Boole & Babbage, Inc., Capex Corporation, CBM, Inc., Cincom Systems, Computer Business Management, Computer Corp. of America, Computer Information Management, Computer Systems & Education, Computer Wares, Inc., Computation Planning Co., Comress, Datamann, Inc., Data Usage Corp., Dynamic Data Services, Inc., Equimatics, Inc./United Systems Intl., Independence Computing & Software Corp., Informatics Tisco, Inc., Information Management Associates, Insurance Systems of America, International Computer Programs, National CSS, Inc., Network Data Processing Corp., Pansophic Systems, Inc., Planning Research Corp., Pryor Computer Software, Systonetics, Inc., System Development Corp., Western Operations, Inc.

code converters involving mathematical procedures of the same kind as those at issue in this case. Two representative examples of such hardware code-converter patents[5] recently issued to IBM and Honeywell (copies attached in the Appendix herein) demonstrate that every one of the arguments put forward by Petitioner's and the supporting hardware manufacturers' briefs against the patentability of the Benson and Tabbot code converter (which arguments ADAPSO/AISC refutes) apply in the same way to patent claims in these IBM and Honeywell hardware patents.[6]

BEMA's Brief (p. 7, f.n. 9) recognizes that its arguments endanger the validity of hardware patents such as the attached ones of IBM and Honeywell, and tries to suggest that distinctions may exist. However, it is quite clear that the hardware manufacturers are prepared to sacrifice their hardware patents if necessary to prevent potential competitors from obtaining software patents.

A patent for a software code converter is not the issue before the Court, except nominally. Such code converters are more typical of hardware patents (witness the attached patents of IBM and Honeywell). A patent for Benson and Tabbot's invention is of negligible importance; it is hardly representative of the grand sweep of the software technology.

The attack of the hardware manufacturers deals with far more momentous matters. They attack hope of innovation and competition in standarized software products so sorely needed by computer users; they attack the very existence of an independent software subindustry within the computer industry; they attack the hope that the computer will help the nation solve some of the awesome problems that it faces.

---

5. IBM Patent U.S. 3,624,637; Honeywell Patent U.S. 3,505,675.

6. Such code converters are typically implemented by hardware. ADAPSO/AISC members would not view such a code converter as representative of their software products.

In order to describe the business stakes involved in this matter, for both the hardware and software companies, as well as the broader interests of the nation, ADAPSO/AISC presents the following outline of the software sub-industry and its relation to the larger computer industry.

NATURE OF THE SOFTWARE SUB-INDUSTRY

Some of the ADAPSO/AISC members market software in competition with hardware manufacturers such as IBM and Honeywell. At least they attempt to do so in a unique business situation which makes it extraordinarily difficult for the software companies to compete with IBM and the other hardware manufacturers.

It is attractive to small business to try to enter and innovate in the software industry because software technology is of such a special character. That is, it is relatively easy and inexpensive to reproduce in the form of a deck of punch cards or a tape recording. Software (what is commonly called a computer program) takes the physical form of a recording of complex combinations of electrical signals whose effect is to structure[7] general-purpose hardware into a unique special-purpose computer. Computer programs can also be built in a hardware form[8] (e.g. a read-only memory), in which case a complete set of electronic switches is replaced each time a program is changed. With the software form, the switches remain in the general-purpose hardware and they are configured by the software signals.

---

7. Ivan Flores, "Computer Software," Science and Technology, May 1969, pp. 16, 17.

8. See IBM's memoranda describing IBM marketing strategy "for putting programming into hardware features (read-only storage) and repricing the present hardware and new programming features," Applied Data Research v. IBM, Civ. No. 158 (D. Minn., filed March 1969) (hereinafter ADR), settled out of court, Affidavit of Martin Goetz in Support of ADR's Motion for Injunction, at p. 22, ¶40.

On the other hand, this ease of reproduction of software has enabled the hardware manufacturers to develop monopoly positions with respect to the software that they market for their own hardware by giving the software away "free"—they actually include the price of their software in the hardware price[9] (an unlawful tie-in practice known in the industry as "bundling"). Monopoly results because "no customer is likely to pay [a software company] for his product if he can get it free from IBM."[10] Under the circumstances, IBM has found that it can make greater software profits by putting the price of the software in the hardware price and saying that the software is "free," "in addition to the obvious marketing advantages of the package."[11] To justify the almost total foreclosure of competition achieved by bundling, IBM has had to devise a false factual picture of the software, calling it a "mental process," which view is contradicted by the line of C.C.P.A. decisions[12] that have found software to be a machine device.

A small software company which is interested in developing a software product and in marketing it is confronted by two major problems of competition. First, the software company must set a price for its product to survive, but this is often difficult, if not impossible, because it may have to compete with the "free" software of

9. Hearing on Automatic Data Processing Before Brooks Subcommittee of the Committee on Government Operations, H. R., 92nd Congress, May 20, 1971, Testimony of D.A. Ink, Office of Management and Budget, pp. 25, 32.

10. Finding of court in Programmatics, Inc. v. IBM, Civ. No. 159 (D. Minn., filed 1969) (hereinafter PI) settled out of court, Transcript of Proceedings on Motion for Preliminary Injunction, at p. 873.

11. IBM Memorandum to A. K. Watson in ADR, *supra* note 8, at p. 23, ¶43.

12. *In re Prater*, 415 F.2d 1393 (1969); *In re Bernhart*, 417 F.2d 1395 (1969); *In re Mahony*, 421 F.2d 742 (1970); *In re Musgrave*, 431 F.2d 882 (1970); *In re Foster*, 438 F.2d 1011 (1971); *In re Benson*, 441 F.2d 682 (1971); *In re Waldbaum*, 173 USPQ 430 (1972).

IBM. Second, if the software product is sufficiently good to be successful, the small company then faces the devastating prospect that, in the absence of patent protection for its product, IBM and the other hardware manufacturers will be able to copy the technological concepts of the priced software and put out their own version, again "free." Thus the very features of software that attract small business to innovate also generally deter software companies from quixotically engaging in competition against "free" software in the absence of patent protection.[13]

For these reasons, the computer industry is twice dominated by IBM. First, IBM has had for years about 75% of the "main frame"[14] business, more than ten times that of its nearest competitor, Honeywell.[15] The remainder of that business is shared by such giant corporations as Sperry Rand, Burroughs, Control Data, National Cash Register; companies such as General Electric and RCA were unable to compete with IBM and were forced to retire from the business.[16]

---

13. See SOFTWARE, a corporate planning report, International Data Corp., Feb. 28, 1972, at p. C-6.

One instance of a software product marketed in competition with a "free" IBM product was the basis of the PI case, supra note 10. (Transcript 775-77). PI offered a sorting product at a price in competition with a "free" IBM sorting product which had never faced any previous competition for the years it was on the market. The price was justified on an advertised claim that PI's sorting product would operate twice as fast and thereby cut IBM's rental charges in half for a major computer task. IBM then marketed a different sorting product, also "free" and operating twice as fast as the previous one, and PI's marketing effort was effectively crippled.

14. "main frame" is the processor or heart of general-purpose hardware that software structures into a special-purpose computer.

15. See Report, Brooks Subcommittee on H. R. 4845, Aug. 17, 1965, 89th Congress, 1st Session, House Report No. 802, "Domination of ADP Industry by IBM," at p. 11; ADR, supra note 8, Goetz Affidavit at p. 11, ¶26, 27; Letter of Richard Sprague to Judge Philip Neville re IBM Discovery in the antitrust suits of U.S. v. IBM and Control Data v. IBM published in Datamation, July 1972, p. 134.

16. See Sprague letter, *supra* note 15, at pp. 134, 135.

The second industry domination by IBM is in software. IBM continues to "bundle" its new system software, as well as all of its older software; that is, IBM saturates the market of users of IBM hardware with all that software as though "free" and forecloses it to competition. In addition, since 1970, IBM has "unbundled" certain of its new software, which it designates as "program products," and which amounts to a trifle of its total software line. Honeywell and Burroughs are completely bundled; other hardware manufacturers also bundle their systems software, and some separately price other parts of their software.[17] IBM's line of "free" software is many times greater than that of any of the other hardware manufacturers.

Thus, IBM's share of the software product market, primarily in the form of the "free" type, is much greater than its share of the "main frame" market. Its share is also affected by the "free" marketing practice which is so effective in foreclosing competition, and by its flexibility in setting prices for those of its products that are priced. IBM's share of that portion of the software market to which it supplies software products is undoubtedly greater than 90%.[18]

The state of affairs of a totally dominated computer industry is aptly summarized in a recent editorial in a computer trade journal[19] entitled "Should One Corporation Have Control Over an Instrument That Will Transform Global Civilization?" (copy attached in the Appendix, p. A-13)

IBM's dominance of the computer industry, and particularly of standardized software products having wide application, has produced the present condition of "poor quality software,"[20] the prime "crucial computer prob-

17. See SOFTWARE, *supra* note 13, at p. A-7.
18. See ADR, *supra* note 8, Goetz Affidavit at pp. 11-13.
19. COMPUTER DECISIONS, March 1972, p. 48.
20. Statement of Dr. Ruth M. Davis before the Government Activities Subcommittee, House Committee on Government Operations, May 16, 1972, at p. 12.

lem" confronting the nation today. As the Director of the Center for Computer Services and Technology of the National Bureau of Standards recently testified before Congress on the "seriousness of the problems besetting both the consumers of computer products and services and the suppliers of these products and services":[21]

> The *crippling problems* are lining up in the software rather than in the hardware field. In particular, they fall under what we call the application-independent problem category. . . . These are the software problems which are common to many applications. (*Emphasis added*)

Small technological companies and independent inventors have been responsible for much of the innovation of this age.[22] A special characteristic of the software technology is that computer scientists working as a small group have been found to be optimally effective in developing new complex software systems.[23] There are approximately 1100 companies, mostly "young and struggling" who "surely possess excellent technical innovative capabilities" and "the principal technical capability . . . to work . . . on our critical computer problems."[24] Because of other special characteristics

---

21. Ibid. at pp. 11 and 12.

22. "Studies of the sources of invention and innovation have shown that independent inventors and small technologically based companies are responsible for a remarkably high percentage of the important inventions and innovation of this century." National Bureau of Standards Experimental Technology Incentives Program, Preliminary Program Description, Fiscal Year 1973 Budget Submission, at p. 9.

23. ". . . all the best programs there are today have been produced by small groups of very able people who were left to get on with it for a year or so in peace. Every man understood everything in the program. A group might grow to a dozen before it lost its coherence. Some big groups have written some very bad programs . . . It seems to be impossible for a very large group to work really efficiently." Lord Bowden, *The Language of Computers*, 58 American Scientist 43, 50, Jan.-Feb. 1970.

24. Davis, *supra* note 20, at p. 15.

of the technology, it offers continuing opportunities for small innovative companies to enter the industry.[24a] Without appropriate incentives and opportunities for such small software companies to compete, this "young and struggling industry"[25] will not grow but will stagnate as a dominated industry. "Imaginative use of computers may well hold the key to necessary improved productivity in our domestic economy,"[25a] and new software concepts can meet this challenge.

Another feature of the software technology of considerable significance to the possibility of small software companies competing with IBM,[26] though the giant hardware manufacturers cannot, is that it enables the building of special-purpose computers (of an endless variety and of great complexity) without requiring a different wiring configuration for each such computer. In place of the "hard wire" of special-purpose hardware, the software uses the "soft-wire"[27] of recorded electrical signals which have the physical effect when placed in the general-purpose computer of setting thousands (or even millions) of electronic switches in unique combinations. Each unique combination of set switches restructures[28] the general-purpose computer to a special-purpose computer. The general-purpose computer can be restructured by software as often and in as many ways

24a. Testimony of M. Jacobs in Hearings on S. 597 Copyright Bill, Before the Subcommittee on Patents, Trademarks and Copyrights of the Senate Committee on the Judiciary, 90th Congress, 1st Session, Pt. 3, at pp. 765-67, 772, 774 (1967).

25. Statement of Dr. Lawrence M. Kushner, Acting Director, National Bureau of Standards, before the Government Activities Subcommittee House Committee on Government Operations, May 16, 1972, at p. 4.

25a. Davis, *supra* note 20, at p. 15.

26. Testimony of M. Jacobs, *supra* note 24a.

27. The term "soft-wire" is used by engineers in the industry to indicate that the recorded signal combinations of software achieve the same effects as actual "hard-wire," but the advantages of modification and replacement without rewiring are also achieved.

28. Flores, *supra* note 7.

as desired, and in an instant, without the slightest change in its hard-wire constructions.

The general-purpose hardware is useless without software, and the special-purpose computers structured by the software is the reason that we have computers and a computer industry.[29] This technological feature of being able to build a new computer system whenever it is wanted without rewiring it is the principal reason for the success of the general-purpose computer and its widespread use; 1000 special-purpose computers can be attained with a single hardware configuration and 1000 software programs.

Small software companies have successfully competed with IBM in past years on the development of standardized software packages,[30] and they can do so in the future if they have appropriate incentives and an open and competitive industry in which to work. The hardware manufacturers are granted patents for their developments built with hardware programming. The counterpart implementations of similar developments by software programming are likewise deserving of patent protection, and such protection is needed to encourage private capital investment in innovative software by small business. The judicial recognition that software is a machine device also will assist in enforcement of the antitrust laws and outstanding injunctions against tie-ins to open the industry to competition.

---

29. "All of the features that have been designed into digital computers may be considered to be, at least to some extent, reflections of software needs." Rosen, *supra* note 3, at p. 1443.

30. See SOFTWARE, *supra* note 17, at pp. A-5, 7 and C-2 to 6.

SUMMARY OF ARGUMENT

The C.C.P.A. decision herein and its prior decisions of the past four years have found as a fact that software, by the nature of the computer technology, is a part of a machine, albeit, as is true of other modern technologies, we cannot visually see its restructuring of the hardware. Since software programming is the engineering equivalent of hardware programming, computer developments in principle and often in practice can be implemented in either machine form, software or hardware. Just as patent protection has been granted to the hardware manufacturers for BCD-to-binary converters and other code converters, and for other programming devices implemented in hardware, so should such protection be granted to software inventors for their programming devices implemented in software.

Machines and their devices and parts have been patentable subject matter from the time of the first patent statutes. Patentability of the software type of machine requires no new principles of law.

The U.S. Patent Office has recognized the need for common search and classification facilities for both forms of technology, and has provided adequate facilities for both.

The only legal protection available for the innovative contributions in software technology is via the patent system. Copyrights and registration proposals would not provide the needed incentive for invention.

The antitrust law implication of the C.C.P.A. finding of fact that software is a machine device is that the bundling of "free" software with hardware by IBM and the other hardware manufacturers is an illegal tie-in in violation of Section 3 of the Clayton Act and Section Act and Section 1 of the Sherman Act.[21] The hardware manufacturers seek to deny the illegal tie-in of one machine device with another, and thus to justify these viola-

tions of the antitrust laws by arguments that the machine device of software is a "mental process" suitable for "free" pricing. These arguments collaterally attack the C.C.P.A. finding of fact that software is a machine device, and fly in the face of the totally machine nature of the computer industry whose very name, "automatic data processing," belies the assertions about "mental processes." Similar false justification is used to explain the Government's failure to enforce the antitrust laws against the tie-ins by hardware manufacturers and the failure to enforce outstanding tie-in injunctions against IBM.

ARGUMENT

I. Software as a machine device is patentable equally with hardware.

The import of Petitioner's position is that general-purpose hardware is a machine, but programmed hardware merely involves the addition of something that Petitioner and the supporting hardware manufacturers call a "mental process" (or a sequence of "instructions" as though it were an instruction manual, or something even more vague and indefinable called "know-how"[32]). In fact, software is actually a form of machine device which structures the general-purpose hardware.

The software form of computer programs is that which is more generally identified as a "program," while the hardware form of computer sequencing is equally a program though less usually so identified since it is in the form of electrical circuits. For example, a recent technical paper[33] by the Director of the Computing Center of Purdue University points out that "much of the sequencing that is done by hardware is the same kind of thing that is done in [software] programming." The paper uses "the term *hardware programming* to indicate programming built into hardware, either through a microprogramming system or by other hardware sequencing methods" (emphasis ours). It uses another term, "extended machine," to characterize the "combined hardware and software system" that is produced when a general-purpose computer is operated under the control of a software program.

---

32. While the hardware manufacturers may not have deprecated software by equating it with mere "know-how" suitable for a give-away, the Government has; see Antitrust Division's Complaint in U.S. v. IBM, Civ. No. 69-200 (S.D.N.Y. filed Jan. 1969, in D. Minn. for discovery).

33. Rosen, *supra* note 3.

This paper[34] and others confirm the well established technological ". . . principle of the equivalence of programming and hardware. This principle begins to invade our consciousness when we recognize that a program for a general-purpose machine turns it into a special-purpose machine."[35]

Thus, the addition of a software program to control a general-purpose computer (thereby creating the "extended machine") is just as much a machine addition to the general-purpose computer as is the additional hardware programming that may be used to enlarge or enhance the character of an existing hardware computer. In both cases, the addition of programming results in a machine system which is different from the machine without the additional programming.

---

34. Ibid. at p. 1443:

"Perhaps the most important part of the job of computer system design is to determine which functions are best handled by hardware, and which should be provided by software. All of the traditional engineering considerations come into play here. One must consider the added cost of additional hardware measured against possible gains in operating efficiency. One must consider the availability of manpower with the appropriate skills for design and for implementation. One must also consider intangible factors such as the inherent advantages of simplicity and the hidden costs of complexity.

"From the point of view of the computer user it makes no difference which features are incorporated into the hardware and which are programmed in software.

"It is the combined hardware-software system, the 'extended machine' that the user must rely on to provide the power and flexibility and convenience that have been promised with each new generation of computing equipment."

35. Continuing the quotation: "The principle becomes more tangible when we compare in detail the logical design of that special-purpose machine with the flow chart of that program, albeit the descriptive language of the logical design is somewhat different from the command language of the program." Final Report No. AD 59 URI, at p. 4, from Univ. of Penna. Moore School of Elec. Eng. by Prof. Saul Gorn, to U.S. Army Signal Research & Development Lab.

The C.C.P.A. found this fact as follows:

> . . . a general purpose computer may be regarded as but a storeroom of parts and/or electrical components. But once a program has been introduced, the general-purpose digital computer becomes a special-purpose digital computer (i.e., a specific electrical circuit . . .)[36]

and again, as follows:

> [W]e say that if a machine is programmed in a certain new and unobvious way, it is physically different from the machine without that program; its memory elements are differently arranged. The fact that these physical changes are invisible to the eye should not tempt us to conclude that the machine has not been changed.[37]

The hardware manufacturers have been obtaining patents for their hardware-implemented investions which were held to be patentable subject matter by the Patent Office Board of Appeals.[38] Moreover, the Patent Office has granted patent coverage to the machine process of hardware-embodied inventions on the theory that the software embodiment is the equivalent of the hardware and the inventor is entitled to protect both forms given his hardware embodiment.[39] The illustrative IBM and Honeywell patents (in the Appendix) both of which have machine and method claims to code converters, are illustrative of this general practice in the Patent Office.

There is no reason why software technology should not be entitled to the same patent treatment as hardware

---

36. *In re Prater*, *supra* note 12.
37. *In re Bernhart*, *supra* note 12.
38. *Ex parte King and Barton*, 1964, 146 USPQ 590. This holding was reached though the Examiner took note of the engineering equivalence of hardware implemented inventions and those implemented by software and general-purpose hardware.
39. *Ex parte Lewin*, 1966, 154 USPQ 487.

programming, they are both machine devices, and that is the substance of the C.C.P.A. decision in this case. Disparate patent treatment of equivalent engineering forms of technology would be a violation of due process.[40] But, it has been achieved for many years by IBM and other hardware manufacturers through their strategy of "free" software and their deprecation, from vaunted positions of technical proficiency, of a technology that has been described by a computer pioneer as "one of the most remarkable of all the intellectual achievements of mankind."[41]

II. The patent office search and classification facilities meet the same standards of adequacy for software as those for the equivalent hardware.

No objections have been raised by Petitioner or the hardware manufacturers to the adequacy of the search facilities for hardware-implemented inventions. The hardware manufacturers know that much of the innovation in hardware was and still is first achieved in software. When a particular innovation was fully and widely accepted, the manufacturers would find it warranted to invest in the mass-production tooling required for hardware and would replace the software with hardware to obtain the advantages of that technological approach.[42]

---

40. The only justification for this difference in treatment was a false and fearsome forecast from the hardware manufacturers that the Patent Office would be deluged with software patent applications if the floodgates were opened. Commissioner Brenner, *"The Future of Computer Programs in the United States Patent Office,"* Law of Software, 1968, pp. B-1, 12. Needless to say, the flood has not materialized in the years since the first C.C.P.A. decision of *Prater* finding software patentable as a machine and machine process.

41. *Bowden, supra* note 23, at p. 46.

42. One danger for hardware manufacturers in software patents is that they may lose full control of development of the hard-

Thus, the hardware manufacturers know, just as does the Patent Office, that the software prior art is as essential as the hardware prior art to adequate examination of the novelty of a hardware-implemented invention. Similarly, both are needed for examination of an equivalent software-implemented invention.

The comments on this point of the Report of the President's Commission on Patents, it is believed,[43] are based on inadequate information supplied to the Commission on the engineering equivalence of hardware and software and the need for corresponding search files and classification facilities for both software and hardware forms of computer inventions.

The issued patents of IBM and Burroughs attached to the Appendix herein demonstrate that adequate search and classification facilities exist in the Patent Office for BCD-to-binary converters and other code converters.

III. Copyrights and legislation for other forms of legal protection of proprietary rights are not effective to protect the machine devices of software.

1. **Other Legislation.** Attempts by IBM and other hardware manufacturers to obtain legislation (35 USC 106 of the proposed Patent Reform Act of 1967)[44] to outlaw patents for software-implemented inventions failed in the Congress. Though the legislation was nominally part of the Administration Bill and had apparently been suggested by the Report of the President's Commission on the Patent System, Assistant Attorney General Turner in

---

ware technology if a larger group of individual inventors and small companies working in the vanguard of the art can obtain such patents.

43. Requests for access to the proceedings of the Commission to investigate these matters were denied by the Patent Office when the Report was published.

44. H. R. 5924 and S. 1042, 90th Cong. 1st Sess. (1967).

charge of the Antitrust Division withdrew all support for 35 USC 106.[45] The President's Science Adviser Hornig likewise withdrew support and gently discredited his own Administration's proposal, testifying: "It has become clear that there are some very genuine and difficult questions of philosophy as well as practicability involved in the patentability of computer programs."[46] The Assistant Attorney General's and the Science Adviser's testimonies followed broad-based hearings held by the Science Adviser to obtain the views of software as well as hardware manufacturers, unlike the President's Commission which did not receive the views of the software manufacturers though it did receive the views of IBM, one of whose vice presidents was a member of the Commission.

With only the Patent Office supporting the proposed 35 USC 106, it was withdrawn from all subsequent bills for reform of the patent system. Though rebuffed in the Congress, the present Petitioner and the hardware manufacturers again raise the same false factual issue of "mental processes" before the Court.

2. Copyrights. The spurious suggestion of the hardware manufacturers that software is a "writing" concerned with "language" can be best appreciated by noting that everything they say about software in this regard is true of their hardware, for the two are engineering equivalents. Just as software is concerned with "language" (for how else does a machine process or communicate data or information except by some "language"), so is hardware concerned with a "descriptive language."[47] Just as a

45. Turner's testimony was silent on Section 106, and did not carry forward any of the views set down prematurely and onesidedly in his 1966 letter quoted in BEMA's Brief (p. 11); see Hearings on H. R. 5924 Before Subcommittee No. 3 of House Committee on the Judiciary, Apr. 20, 1967, at p. 135.

46. Hearings on H. R. 5924, Ibid. at p. 160.

47. Namely, "the descriptive language of the logical design" of the hardware, *supra* note 35. For the relation of the "command language" of software programming and the logical design language of hardware programming, see Jacobs, *supra* note 2, at p. 52.

computer scientist "writes" with pencil and paper in setting down his software concepts, so he does in setting down the logic design of his equivalent hardware. Just as it would make no sense to copyright hardware (except as one wanted to protect drawings or written descriptions of it) so it makes no sense to copyright the equivalent software (other than drawings or written descriptions). Were this matter to be before the Court on a copyright infringement charge based on the infringer's use of software to direct the operation of hardware, the outlandishness of the situation would compel the question: "How can you possibly enjoin the use of a machine device or process by copyright?" The answer is: "You cannot!"[48]

The hardware manufacturers' proposal that one protect software by copyrights or by some other new statutory approach reflects their lack of interest in inventive innovation, but instead their interest in "non-inventive innovations."[49] As the Court has stated numerous times, only the exacting standards of novelty required by the Patent System to meet the constitutional standard of invention afford a basis for obtaining exclusive rights in machines and articles of manufacture.[50]

---

48. The hardware manufacturers, it seems, do not actually state that copyright protection extends to use of machine device. But they do suggest that copyright protection of written records of software would somehow be meaningful protection. See *Jacobs, supra* note 2, at pp. 59, 60.

49. See Galbi, Proposal for New Legislation to Protect Computer Programming," 17 Bulletin Copyright Society of the U.S.A. 280, 281.

50. See *Graham v. Deere* (1966) 383 U.S. 1, reaffirming *Hotchkiss* (1850) 11 How. 248; *Sears v. Stiffel* (1964) 374 U.S. 826; *Mazur v. Stein*, 98 L.Ed. 630, 642. See Testimony of Jacobs, *supra* note 26, at p. 776.

IV. The finding of fact that software is a machine device and patentable will encourage innovation and competition in the computer industry and will assist in eliminating antitrust violations and their anti-competitive efforts.

The C.C.P.A. decisions[51] have found as a fact that the hardware and software combination of a programmed digital computer forms a special-purpose computer whose character is determined by the software. These decisions, in effect, acknowledge the facts of this industry, namely, that the software is the reason for buying computers. That is, the various machines structured by software out of the elementary operations of the hardware are the *raison d'etre* of the hardware. This C.C.P.A. finding of fact is a recognition of the software manufacturer's important place in the industry. By recognizing software for what it is, namely, a machine device and the control mechanism that structures a special-purpose computer system, the C.C.P.A. found no problem in granting patent protection under existing statutes. By this line of decisions, small business was given an incentive to invest in innovation.

A hardware manufacturer's prime incentives are to increase the size and quantity of the hardware that it prices and sells, and to increase the hours of use of the hardware that it prices and leases on an hourly rate. Such a manufacturer does not have an incentive to improve the quality or efficiency of the "free" software, such as may result in a smaller size, faster operation, or more efficient use of the hardware and, thereby, in reduced revenues.[52] In this context, the hardware manufacturers find it in their interest to maintain their present control of the software markets for their respective hardware, and

51. In re *Prater*, *supra* note 12; in re *Bernhart*, *supra* note 12.

52. See *PI v. IBM*, *supra* note 10; IBM did not place an improved sorting program on the market until the PI sorting program was marketed with double the operating speed.

to oppose patents for software that would offer vehicles for entering competition.

As builders and sellers of machines and machine devices, the software manufacturers are in direct competition with the hardware manufacturers for the software market, a billion dollar market [if we include the "free" IBM software] and potentially the principal market of the computer industry. Therefore, with acceptance of the fact that software is a machine device and not a mental process, the bundling of "free" software and priced hardware is an illegal tie-in.[53]

The user of computers buys or leases a computer system on the basis of the particular automatic data processing tasks that have to be performed. Software packages structure the hardware to provide the customer with special-purpose computers to perform those tasks. The hardware merely supplies a repertoire of basic operations ("instructions" or "commands"), a certain speed of performance of the operations, and a memory and other equipments for input and output.

Thus the C.C.P.A. finding of fact that the software is a machine device which structures the hardware reveals the distortion in marketing the hardware as the principal computer component, although the principal attractions for the customer are the special-purpose computers created by the software. The C.C.P.A. finding of fact helps to reorient the emphasis—its identification of the software as the structure of the special-purpose machine clarifies that in effect the priced hardware is tied to the "free" software. This is not a new approach; in the 1935 perpetual tie-in injunction against IBM, upheld

---

53. Under Section 3 of the Clayton Act, *Times Picayune v. U.S.*, 345 U.S. 594 (1953); *U.S. v. IBM*, 298 U.S. 131 (1935); *Advanced Business System v. SCM*, 415 F.2d 55 (C.A. 4, 1969).

Under Section 1 of the Sherman Act, *Fortner Enterprises v. U.S. Steel*, 394 U.S. 495 (1969), *No. Pacif. Ry. v. U.S.*, 356 U.S. 1 (1958), *Washington Gas Light Co. v. Virginia El. & P. Co.*, 1970 Trade Cas. para. 73073 (E.D. Va. 1970).

by the Court,[54] IBM is forbidden from giving its tabulating machines away "free" in order to sell its punch cards. The software, which carries the structure of the special-purpose computer desired by the customer, is given away "free" in order to sell the priced general-purpose hardware.

The C.C.P.A. finding of fact that the software-hardware combination is a machine and the software a machine device also offers the basis for a substantial change of these practices that have held back the development of software so long in this industry. That is, the approach promises the necessary vehicle for enforcement of the 1956 IBM *Consent Decree*[55] and injunction against IBM, which updated the 1935 injunction. The 1956 injunction (copy of pertinent portions attached in Appendix, p. A-14) prohibits the tie-in of the different parts ("machine or device and attachments therefor used primarily in or with an electronic data processing system") of an overall computer system. The Antitrust Division of the Justice Department attempts to justify its refusal to enforce that injunction against software bundling by the argument that software does not come within the definition of "machine or device and attachments." The accompanying cry of "mental process" has a hollow ring when it serves as a support for the maintenance of long-standing violations of the antitrust laws.

This C.C.P.A. finding of fact also offers promise of enforcement of Section 3 of the Clayton Act and Section 1 of the Sherman Act, proscribing illegal tie-ins against the hardware manufacturers supporting Petitioner, who are all guilty of the bundling practice in varying degrees,

---

54. *U.S. v. IBM*, Ibid. ¶4b of the decree.

55. *U.S. v. IBM*, 1956 Trade Case, para. 68245, Secs. XV(b) II(f) and II(g) (S.D.N.Y. 1956). In the pertinent portions of the *Consent Decree* set forth in the Appendix, Sec. XV(b) is the applicable injunction and Secs. II(f) and II(g) set forth the principal definitions.

and who come into this Court talking of "mental processes," "mathematics" and "writings" to disguise their anti-competitive and monopolistic practices of bundling and tying of machines and machine devices.[56]

It may be that those giant corporations who are hardware manufacturers and who together have less than 25% of the total main frame business, as against IBM's 75%, need, in order to compete, the assistance of the revenues stemming from monopolies of the software for their own main frame hardware. However, whatever that need, continuing violations of the antitrust laws through bundling cannot be countenanced. Software companies can compete in the open market with IBM, even if the giant hardware manufacturers cannot. Certainly, the closing of the IBM software market to the small independent software companies, which is the result of the failure to enforce the *Consent Decree* against IBM, cannot be justified. Moreover, as the Court said in the 1935 IBM tie-in case, there is "no tenable basis for an exception in favor of a condition whose substantial benefit to [IBM] is the elimination of business competition and the creation of a monopoly."[57]

The importance of this case to a twice-dominated computer industry cannot be tested by the value of a minor code converter patent. Thus, the finding of fact of the C.C.P.A. that software is a machine device should become a first step in opening the software sub-industry to competition. The C.C.P.A. decision that, as a machine device, software is patentable subject matter, makes it possible for small software companies to obtain patents that will aid in curbing the otherwise uncontrolled anti-competitive practices of IBM and other hardware manu-

---

56. Any hardware company obtaining patents for software, since the priced hardware is tied to "free" software, would be faced with the presumption of "sufficient economic power" which is an element of the violation. *International Salt Co. v. U.S.*, 332 U.S. 392 (1947).

57. *U.S. v. IBM*, *supra* note 53, at p. 140.

facturers. Patents for software inventions will serve as an incentive for small software companies to invest in research and development for the innovative software sorely needed by all computer users and by the Government, and as a shield to market a new or improved product in competition with the "free" software of IBM. Whatever monopoly may be created by the exclusive rights of the patent grant, it is a *de jure* monopoly that will tend to limit the *de facto* monopoly that has been countenanced in the software sub-industry for so many years.

CONCLUSION

The findings of fact and conclusions of law of the Court below are soundly based, and its decision should be affirmed.

Respectfully submitted,

MORTON C. JACOBS
828 Suburban Station Building,
Philadelphia, Pa. 19103
Attorney for Amicus Curiae
ADAPSO/AISC

Of Counsel:
McCLURE, MILLMAN & JACOBS
828 Suburban Station Building
Philadelphia, Pa. 19103

FOLDOUTS ARE TOO LARGE TO BE FILMED